YRDKRX63N

Project Generator Guide

R01US0048EU0100
Rev. 1.00
February 24, 2012

Introduction

This document describes the applications created by the Project Generator for HEW included with the YRDKRX63N Demonstration Kit.

Target Device

Renesas YRDK RX63N

Contents

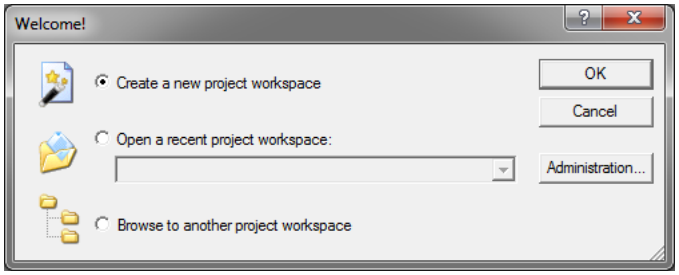
1. Introduction

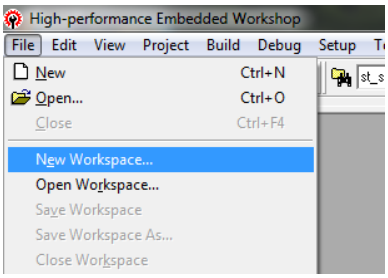
The Renesas High Performance Embedded Workshop (HEW) provides embedded systems developers with a complete, integrated development environment (IDE) dedicate to creating, building, and debugging firmware for Renesas MCU's. HEW has an open architecture that allows tools to be "plugged in" to the base installation to extend functionality and offer new features. This includes the capability to add Project Generators that help automate the creation of new firmware projects. Included on the CD that ships with the Renesas YRDKRX63N Demonstration Kit is a Project Generator that creates projects that demonstrate various peripherals on the RX63N MCU that is on the YRDK board.

This document details how to use the Project Generator, along with a brief description of the projects that the Generator creates.

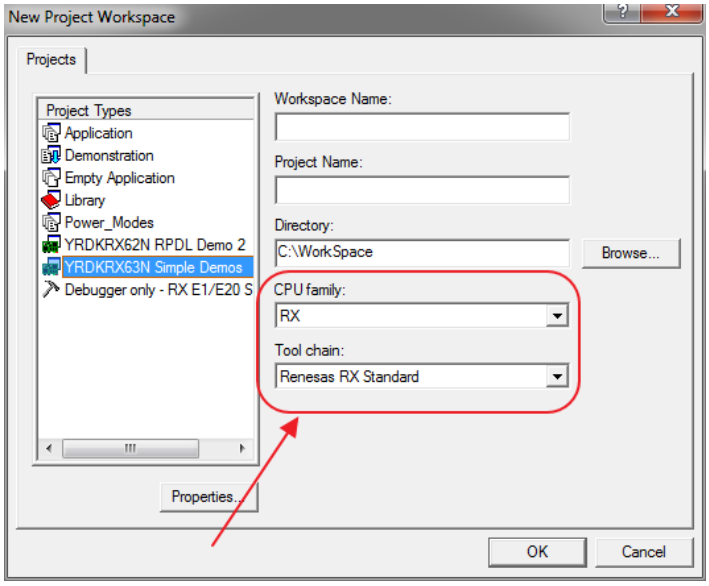
2. Using the Project Generator

To use the Project Generator, start HEW. The welcome screen is displayed. Choose "Create a new project workspace". If you've already bypassed the welcome screen or are in another project, select the "Create new workspace" option from the file menu.

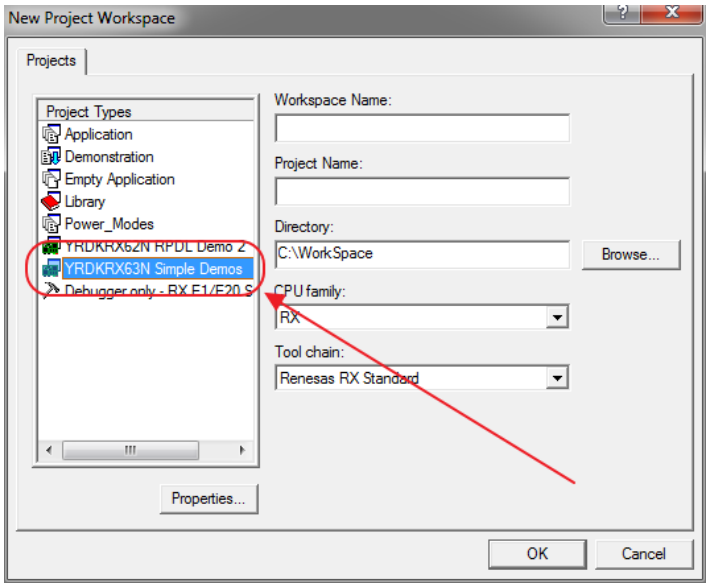




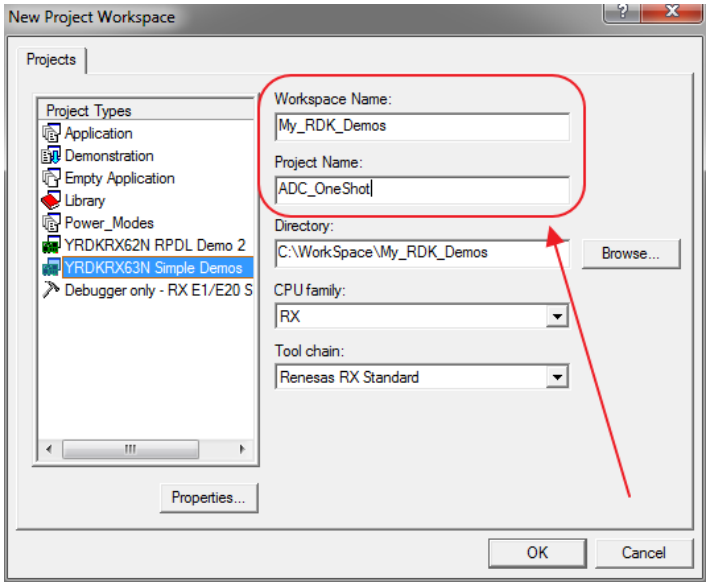
The New Project Workspace dialog is shown. To show the Project Generator for the YRDKRX63N, first select the correct CPU ("RX") and compiler ("Renesas RX Standard") as shown. Your list of Project types may be different, and it changes based on the selected CPU and tool chain.



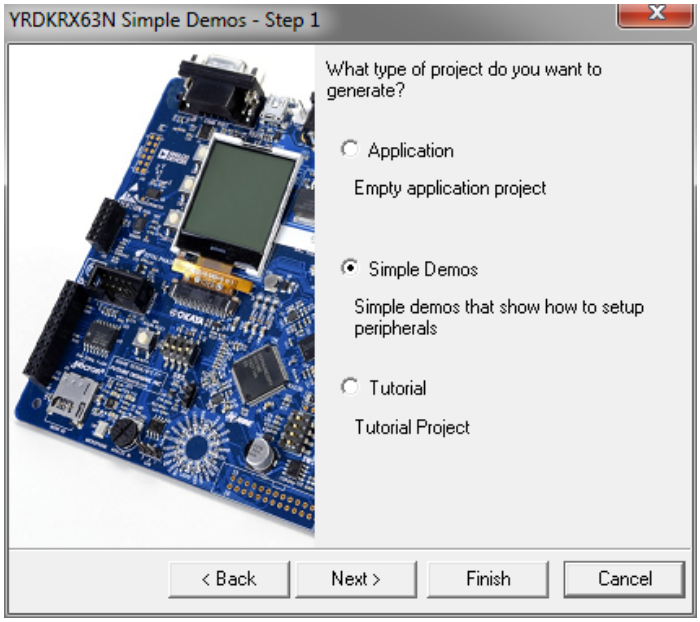
Next, choose “YRDKRX63N” from the list of Project Types on the left:



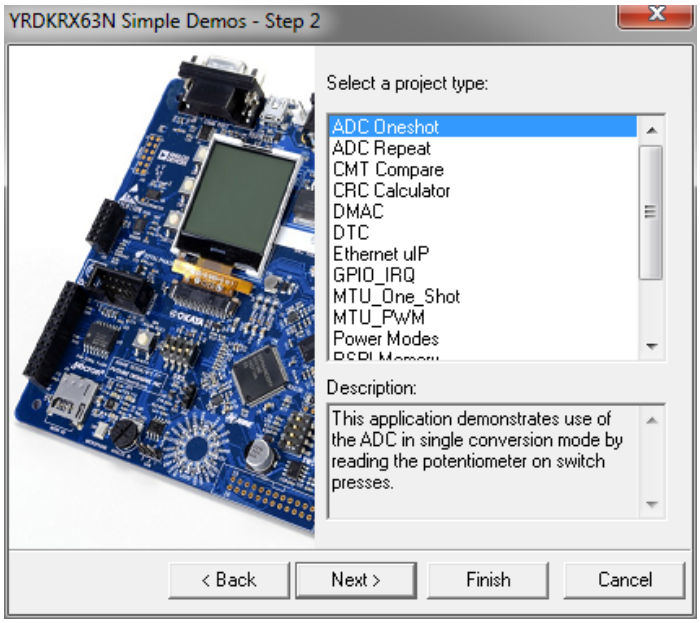
Now give your new workspace a name and give the first project in the workspace a name. An example is shown below:



Click “OK” and the Project Generator starts, displaying a dialog that offers three different types of projects: a tutorial, a sample code project that demonstrates one or more peripherals, or an empty application with not much more than startup code. In this example, a Sample Code project is selected. Click “Next” to continue.

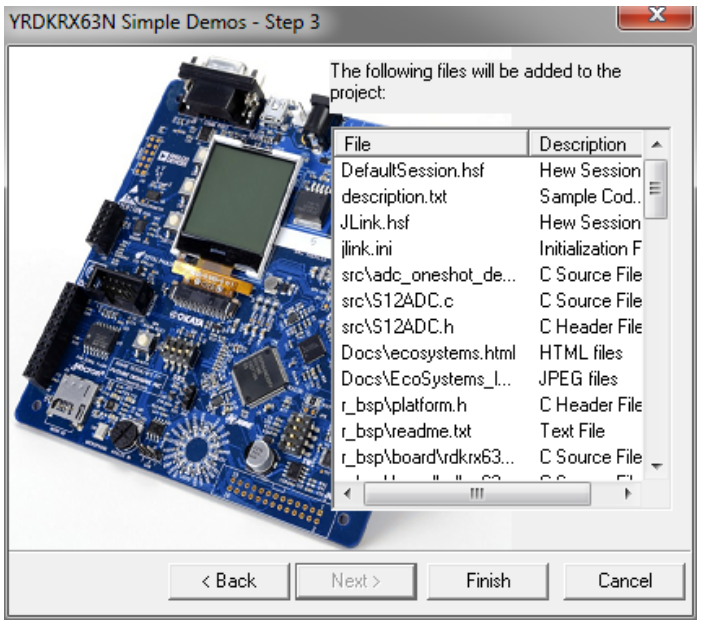


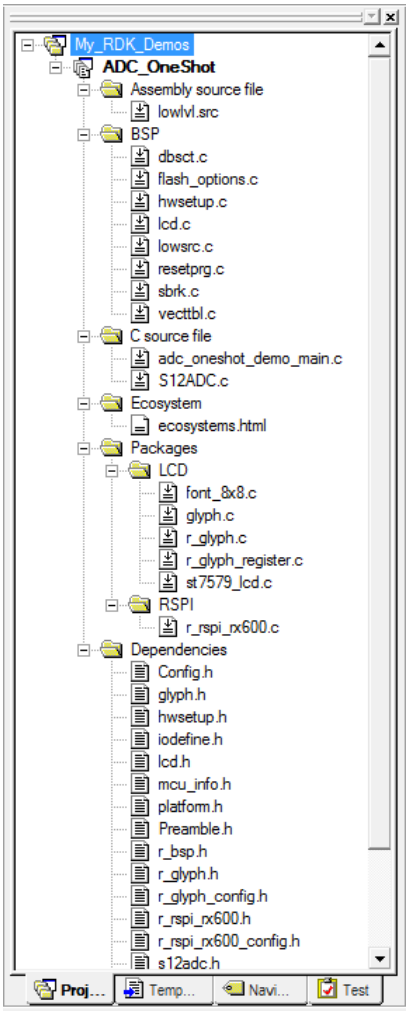
For the Tutorial and Application types there are no other options. With the Sample Code project type, a list of project types is offered. Each type generates a complete running program that demonstrates the use of one or more peripherals. As you highlight each type in the top list, a short description is shown. For this example, we’re going to create a project that shows how to read the ADC in single conversion mode. Click “Next” to continue.



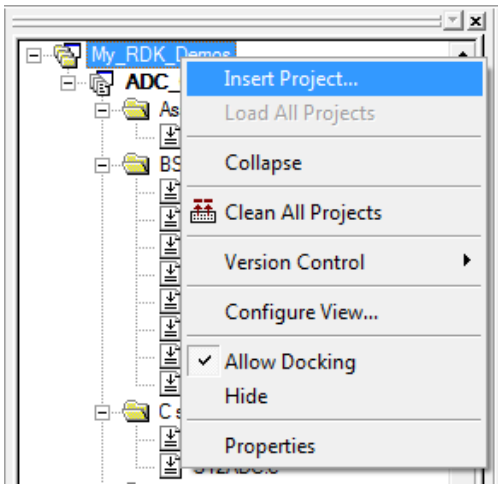
The next dialog shows a list of the files that will be added to your new project. Click “Finish” to continue.

The Project Generator goes to work and creates your new project. The Projects pane in HEW shows the result:





To add more Project Generator code to the same workspace, right click on the workspace name in the Projects pane and choose “Insert New Project…” Repeat the steps above to create the new project.



3. Description of Generated Applications

3.1 Application

This is an empty project with just a “while” loop in the main function. The project demonstrates basic hardware initialization. Review the file HWSetup.C to see how the RDK hardware is initialized. Only basic setup is performed such as setting the direction of the I/O pins.

3.2 ADC One Shot

This sample demonstrates use of the ADC in single conversion mode. The program takes a sample reading of the voltage created by the potentiometer VR1 and displays it on the LCD.

3.2.1 Operation

1. Build and download the sample code to the YRDK.
2. Click 'Reset Go' to start the software.
3. Instructions are displayed on the LCD. Adjust the potentiometer VR1, and press switch SW1 to trigger an AD conversion.
4. The ADC result will be displayed in both ADC counts and volts.
5. Re-adjust VR1, and press SW1 again to perform another conversion.

3.3 ADC Repeat

This sample demonstrates use of the ADC in continuous scan mode. After the ADC is started, a timer is triggered to periodically fetch the result from the conversion. The potentiometer VR1 is connected to the ADC; adjust it and watch the results on the LCD.

3.3.1 Operation

1. Build and download the sample code to the YRDK.
2. Click 'Reset Go' to start the software.
3. The LCD will show the name of the program and the current ADC value.
4. Adjust the potentiometer, VR1, and observe the values displayed on the LCD change.

3.4 CMT Compare

This sample demonstrates the use of a CMT configured to generate an interrupt every 100 milliseconds. The interrupt handler toggles the user LEDs.

3.4.1 Operation

1. Build and download the sample code to the YRDK.
2. Click 'Reset Go' to start the software.
3. The LED's are toggled every 100ms by the Compare Match Timer. CMT channel 0 is set to generate an interrupt every 100 ms. During the interrupt the LED's are toggled. The CMT is a great choice for this type of periodic task because is automatically clears and re-arms itself, so no further servicing of the CMT is necessary. This makes it suitable for tasks like generating the base tick for a real-time operating system (RTOS) or pacing a simple scheduler.

3.5 CRC Calculator

This sample demonstrates the CRC unit by calculating the checksum of a sequence of values typed in from a terminal.

3.5.1 Operation

1. Build and download the sample code to the YRDK.
2. Connect the YRDK to the host PC via an RS232 serial cable.
3. Launch a suitable terminal program (e.g. HyperTerminal) and configure it to operate on the channel the YRDK is connected to with the following settings:

   Baud.....115200
   Data Bits.....8
   Parity.....None
   Stop Bits.....1
   Flow Control.....None

4. Click 'Reset Go' to start the software.
5. Observe the terminal window - instructions will be displayed at the top, asking the user to input a character.
6. Enter an even number of up to 64 hexadecimal characters (0-9, A-F) into the terminal, and each character will be echoed back to the terminal display. Only hex characters will be accepted and displayed. Press Enter when done.
7. The CRC checksum result will be displayed in hex format.
8. To verify the checksum, re-enter the same exact sequence of characters followed by the CRC result that was displayed. The CRC checksum will be recalculated, and it should show a new result of 0x0000. If you don't get this result, double check to see that you actually entered an even number of digits, because CRC can only be calculated on a complete byte.

3.6 DMAC

This sample demonstrates the DMAC unit by performing continuous DMA transfers from a single source address, the output of the 12-bit ADC, and alternately filling each of a pair of destination buffer arrays with the results.

3.6.1 Operation

1. Build and download the sample code to the YRDK.
2. In the dmac_demo.c module, select the variables 'g_dmac_dest_buf_1' and 'g_dmac_dest_buf_2' and add them to the HEW watch window.
3. Click the + symbols in the watch window to expand these array variables. The contents of the DMAC transfer destination buffers can now be observed in the variable watch window when the program runs. Click the auto update button ("R" icon) to have the contents update while the program is running.
4. Click 'Reset Go' to start the software
5. Observe the DMAC transfer destination buffers being filled with the ADC readings. Turn the potentiometer on the YRDK to see the values changing.

Note: In order to watch the contents of the destination buffers during execution, the sample must be built in 'debug' mode.

3.7 DTC

This sample demonstrates the DTC unit; by using the DTC to transfer a series of ADC conversions to a 16 element array.

3.7.1 Operation

1. Build and download the sample code to the YRDK.
2. Right-click the variable 'g_dtc_destination' and select 'Instant Watch', clicking 'Add' on the subsequent dialog.
3. The contents of the variable 'g_dtc_destination' can be viewed in the watch window. Select 'R' for auto update.
4. Click 'Reset Go' to start the software. Instructions will be displayed on the debug LCD.
5. Adjust the potentiometer, VR1, and press the switch SW1. The result from the ADC conversion will be transferred into the 16 elements of the transfer destination array at timed intervals using a timer interrupt mapped to the DTC.
6. While the example is taking samples, the VR1 pot can be turned to vary the ADC result.
7. Once the last array element has been transferred, the next interrupt from the timer will stop the timer.
8. Press switch SW1 again to repeat the process.

Note: In order to watch the contents of the g_dtc_destination array during execution, the sample must be built in 'debug' mode.

3.8 Ethernet uIP

This sample shows the open-source uIP Ethernet stack ported to the RDK.

3.8.1 Operation

1. Build and download the sample code to the YRDK.
2. Connect the YRDK to your network (be sure there is a DHCP server accessible from the network).
3. Click “Reset Go” so start the software.
4. The YRDK will look for a DHCP server to obtain an IP address. Once an address is obtained, it is displayed on the LCD.
5. Type the IP address into the address bar of your web browser to contact the web server running on the YRDK.

3.9 GPIO and Pin IRQ demonstration

This project demonstrates the use and setting of GPIO ports to read the state of input pins, and to control output pins. In addition, transitions on an input pin can be made to generate an interrupt (IRQ). The YRDK on-board switches are set up as pin IRQs and an LED is set up to be driven on or off through the control of an output pin.

3.9.1 Operation

1. Build and download the sample code to the YRDK.
2. Click 'Reset Go' to start the software. Instructions will be displayed on the LCD.
3. Press and hold switch 1 to observer LED4 turn on and stay on.
4. Release switch 1 to see LED4 turn off. Repeat as many times as desired.
5. Observe that each time the switch is pressed a count is updated to the LCD. Switch presses are counted as “Down” counts and switch releases are counted as “Up” counts. If switch debouncing is working then the number of down counts and up counts should always match when the switch is released. This demo shows that IRQ interrupts can be configured separately for rising and falling edges.
6. Press switch 2 to zero out the switch 1 counts

3.10 MTU One Shot

This project demonstrates the use of the MTU timer to create a single event time interval count that triggers an interrupt at a preset count match.

3.10.1 Operation

1. Build and download the sample code to the YRDK.
2. Click “Reset Go” to start the software.
3. Press SW1 to start a one second delay.
4. Observe the red LEDs illuminate.
5. After 1 second the red LEDs are turned off and the green LEDs light up. Repeat as many times as desired.

3.11 MTU PWM

This sample application sets up a MTU channel to generate a pulse-width modulated waveform on an output pin that directly drives an LED. In addition, an interrupt service routine changes the PWM duty cycle and toggles the port pins that drive more LEDs. The brightness of the LEDs is ramped up and down as the duty cycle changes.

3.11.1 Operation

1. Compile and download the sample code. Click “Reset Go” to start the program.
2. Observe LEDs alternately brightening and dimming by means of PWM switching. The PWM duty cycle is ramped up and down continuously.
3. The PWM waveform can be measured at JN2 pin 23 with an oscilloscope.
4. LED9 is switched directly by the PWM output pin from the MTU. The remaining LEDs are all switched by port command during the MTU compare-match interrupt. Note that the duty cycle has been intentionally inverted between LED9 and the rest so that it is easily distinguished from the others.

3.12 Power Modes

This project demonstrates the MCU Sleep and Software Standby low power modes.

3.12.1 Operation

1. Build and download the sample code to the board.
2. Click “Reset Go” to start the program.
3. Observe the green LEDs flashing
4. Press SW1 to place the MCU into Sleep mode, or press SW2 to place it into Software Standby mode. The green LEDs will turn off and the red LEDs will briefly light.
5. After the MCU has gone into either the Sleep or Software Standby mode, press any of the switches SW1 through SW3 to return to normal run mode. The green LEDs will resume flashing.
6. The program repeats after a cycle of steps 2-5.

3.13 RIIC Master

This project demonstrates use of the RIIC unit in master mode by reading the ADXL345 digital accelerometer and the ADT7420 digital thermal sensor (if present).

3.13.1 Operation

1. Build and download the sample code to the YRDK.
2. Click “Reset Go” to start the software. The name of the sample will be displayed on the LCD.
3. Pick up the board and tilt the board forward/backward and left/right.
4. As you slowly roll the board in a circular motion, observe the circle LEDs light to indicate the direction of the tilt.
5. If your board has a temperature sensor installed, the temperature is displayed on the LCD. Blow on the sensor or hold your finger on it to see the temperature change.

3.14 RSPI Memory

Demonstrates use of the RSPI peripheral to write and read from the RSPI attached flash device. This program writes 2048 bytes of data to the flash device located on RSPI-0, 64 bytes at a time. It then proceeds to read the data back 512 bytes at a time and compares it to the original data. Upon successful completion, LED 4 is turned on. If the programming fails, LED 14 is turned on.

3.14.1 Operation

1. Build and download the sample code to the YRDKRX63N.
2. Click “Reset Go” to start the software.
3. Observe messages on the LCD.
4. Observe LEDs for result status.

3.15 SCI Async

This application shows the use of a Serial Communications Interface (SCI) channel to send and receive data over the RS232 interface.

3.15.1 Operation

1. Compile and download the sample code.
2. Connect RS232 port on the board to your PC with a serial cable. You will need a terminal program such as HyperTerminal or TeraTerm. Serial port settings should be set to 115200bps, 8-bit data, no parity, 1 stop bit, and no hardware handshake.
3. Click “Reset Go” to start the software.
4. Observe the LEDs on the board flash.
5. A sign-on message is sent by the board to the terminal program on your PC.
6. Characters you type into the terminal window are echoed back by the board.

3.16 TMR One Shot

This project demonstrates the 8-bit Timer (TMR) peripheral.

3.16.1 Operation

1. Compile and download the sample code.
2. Click “Reset Go” to start the software.
3. Observe LED4 flashing and instructions displayed on the LCD.
4. To start the one-shot time interval, press SW1.
5. Observe the red LEDs lit during the 1-second time interval. The free-running counter for the TMR channel is displayed on the LCD. When the counter matches the compare value, the delay is complete.
6. At the conclusion of the 1-second interval the red LEDs turn off and the green LEDs turn on.
7. Go to step 4 to repeat.

3.17 Tutorial

This tutorial sample will demonstrate basic use of the YRDK hardware and the J-Link debugger.

3.17.1 Operation

1. Build and download the tutorial project to the YRDK.
2. Click 'Reset Go' to start the software.
3. "Renesas RX63N" will be displayed on the debug LCD, and the user LEDs will flash.
4. The user LEDs will flash at a fixed rate until either a switch is pressed, or the LEDs have flashed 200 times.
5. The software will then vary the rate in which the LEDs flash by the position of the potentiometer, VR1. Turn the potentiometer in both directions, and observe the change in flash rate.
6. While the LEDs flash at a varying rate, the second line of the debug LCD will show “STATIC". The second LCD line will slowly be replaced, letter by letter, with the string "TESTTEST".
7. Once the second line of the debug LCD shows "TESTTEST" fully, it will return back to showing "RX63N". The LEDs will continue to flash at a varying rate until the tutorial is stopped.
8. In order to repeat the tutorial, click 'Reset Go'.

3.18 Watchdog Timer

The Watchdog Timer (WDT) can protect systems by resetting the MCU if software fails to periodically service the WDT. The WDT has a count register that continuously decrements when the WDT is armed. When software services the WDT, the count register is reset to its initial value; if the down count register reaches zero the MCU is reset. This project allows you to delay servicing of the WDT to see the effects.

3.18.1 Operation

1. Compile and download the sample code.
2. Click “Reset Go” to start the software.
3. The display shows the value of the WDT down counter.
4. Pressing and holding SW1 forces the software to stop servicing the WDT. While SW1 is held down observe the WDT down counter decrementing towards zero.
5. If you release SW1 before the count reaches zero, the MCU continues running.
6. If SW1 is held down until the down counter reaches 0 the MCU resets and the program restarts. On restart after WDT underflow the status register is read and displayed on the LCD, showing the underflow flag set.

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Feb.24.2012	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1